JUL 2 1 2005
PATENT & TRADEMARK OFFICE

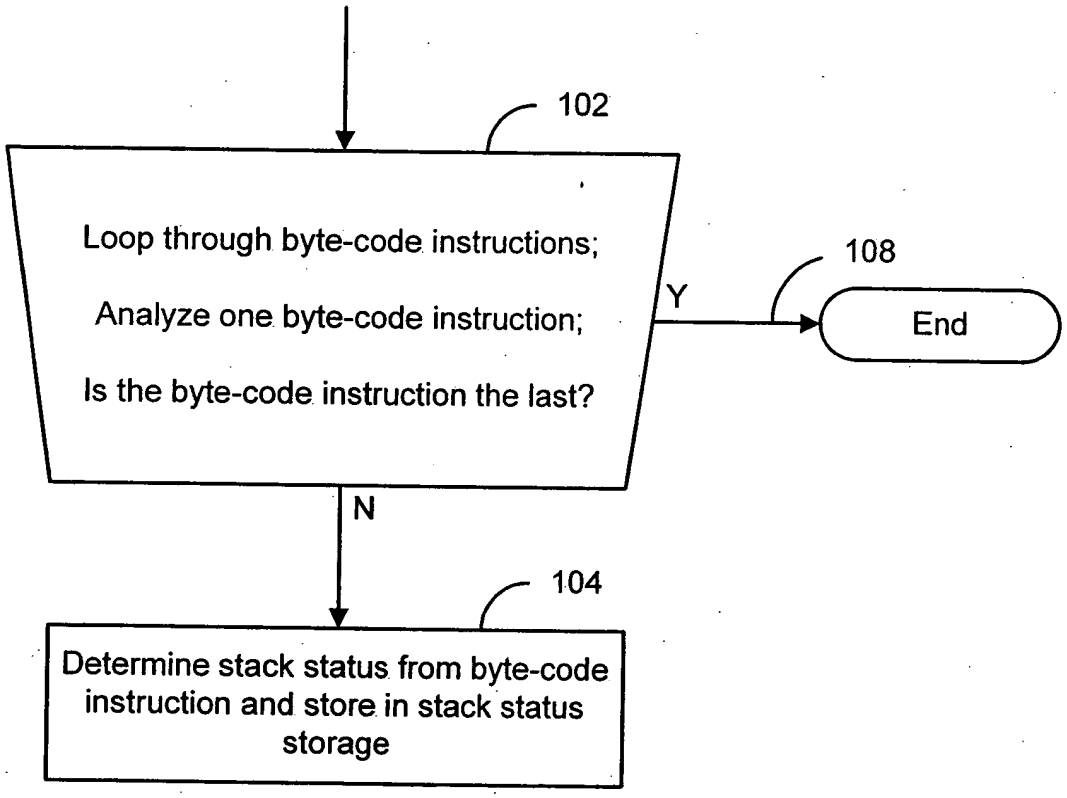

FIG. 1A
(Prior Art)

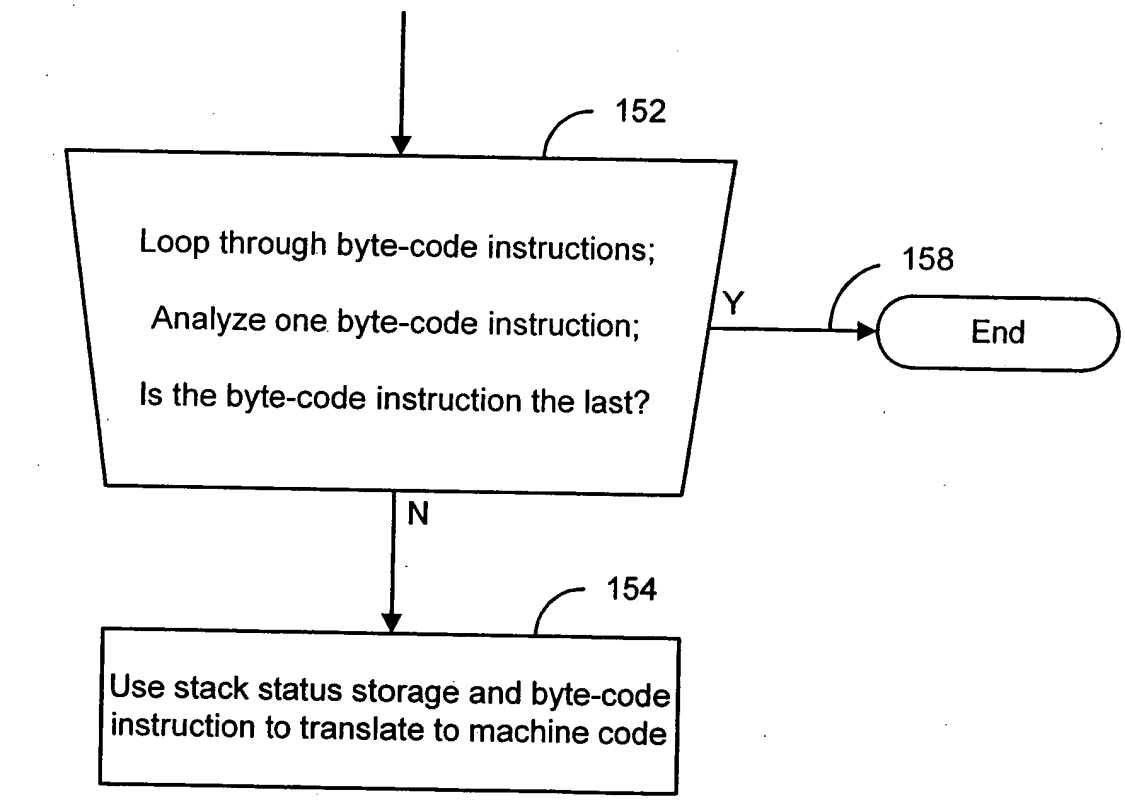

FIG. 1B
(Prior Art)

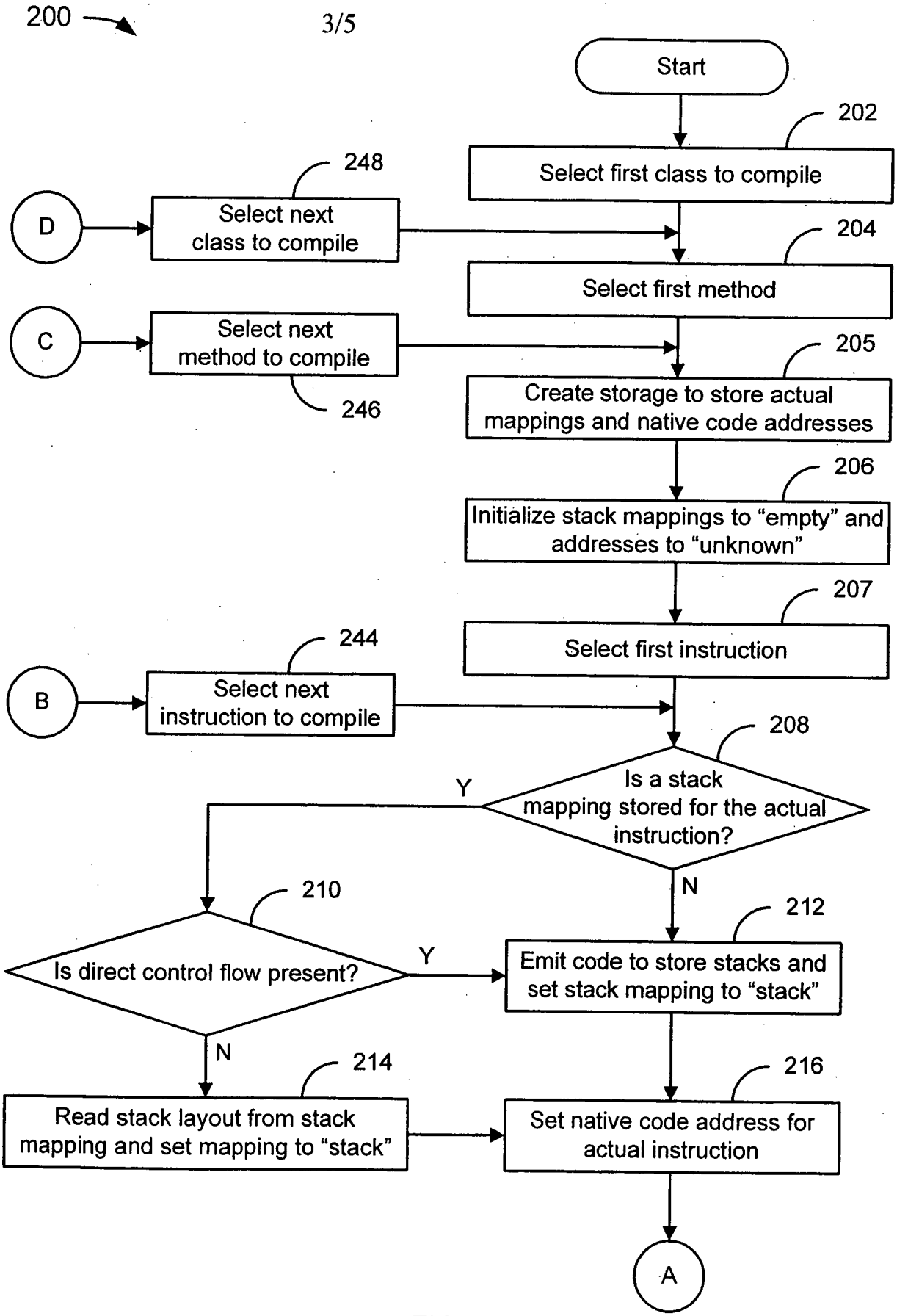
200
Start
202
Select first class to compile
248
D
Select next class to compile
204
Select first method
C
Select next method to compile
246
205
Create storage to store actual mappings and native code addresses
206
Initialize stack mappings to "empty" and addresses to "unknown"
207
Select first instruction
244
B
Select next instruction to compile
208
Is a stack mapping stored for the actual instruction?
Y
N
210
Is direct control flow present?
Y
212
Emit code to store stacks and set stack mapping to "stack"
N
214
Read stack layout from stack mapping and set mapping to "stack"
216
Set native code address for actual instruction
A

FIG. 2A

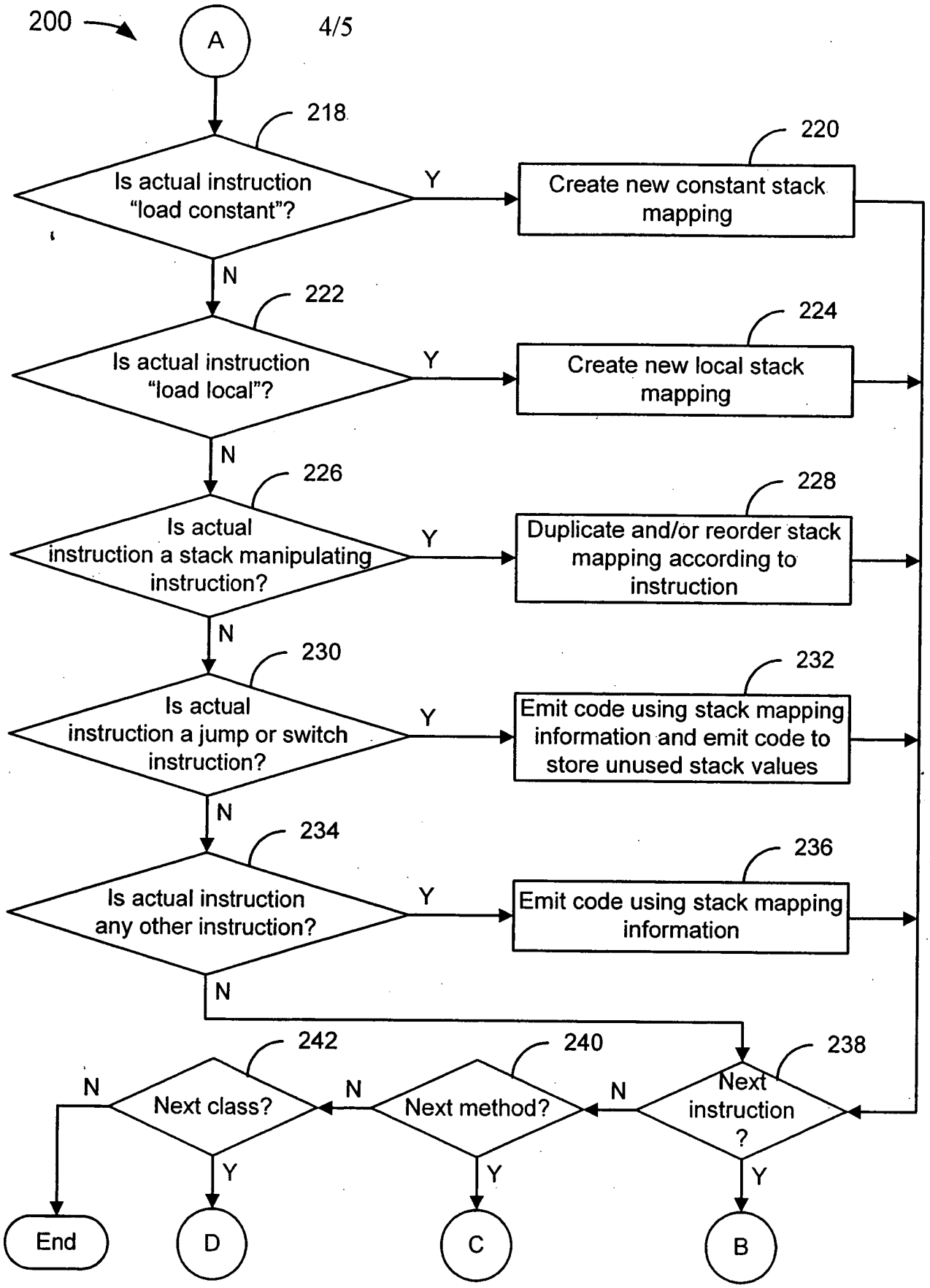
200
A
218
Is actual instruction "load constant"?
Y
220
Create new constant stack mapping
N
222
Is actual instruction "load local"?
Y
224
Create new local stack mapping
N
226
Is actual instruction a stack manipulating instruction?
Y
228
Duplicate and/or reorder stack mapping according to instruction
N
230
Is actual instruction a jump or switch instruction?
Y
232
Emit code using stack mapping information and emit code to store unused stack values
N
234
Is actual instruction any other instruction?
Y
236
Emit code using stack mapping information
N
242
Next class?
N
240
Next method?
N
238
Next instruction ?
N
Y
Y
Y
End
D
C
B

FIG. 2B

300 →

<table>
<tr><td rowspan="5">For each value on the bytecode stack</td><td rowspan="4">A field showing actual mapping to storage in target machine</td><td>Constant</td></tr>
<tr><td>Local</td></tr>
<tr><td>Temporary</td></tr>
<tr><td>Stack</td></tr>
<tr><td rowspan="3">A field containing additional information</td><td>Constant Value</td></tr>
<tr><td>Slot Number</td></tr>
<tr><td>Register Number</td></tr>
<tr><td>For each target of a jump or switch instruction</td><td colspan="2">A field to store native code address</td></tr>
</table>

FIG. 3